IN THE CLAIMS

Please amend the claims as indicated:

1-22. (cancelled)

23. (new) A computer-implemented method of verifying that a reduced set of code adheres to a pre-defined language syntax before being stored in a smart card, the computer-implemented method comprising:

determining if an original full-version code file complies with a pre-defined language syntax, wherein the original full-version code file is capable of executing all instructions and utilizing all features found in a full version of a programming language;

in response to determining that the original full-version code file complies with the pre-defined language syntax, creating a first cryptographic signature for the original full-version code file;

storing the first cryptographic signature on a smart card, wherein the smart card comprises a limited-capacity embedded microcontroller;

converting the original full-version code file into a reduced-version code file, wherein the reduced-version code file is capable of executing only a limited subset of instructions and features found in the full version of the programming language;

converting the reduced-version code file into a converted file, wherein the converted file is capable of utilizing all features found in the full version of the programming language, and wherein converting the reduced-version code file into the converted file is performed by a pre-conversion step, a mapping step, and a final conversion step,

wherein the pre-conversion step includes:

converting the reduced-version code file into a preconverted file, wherein the preconverted file utilizes the pre-defined language syntax, and wherein the preconverted file includes a preconverted code section and a preconverted code description section, and

wherein the mapping step includes:

utilizing a mapper to replace, in the preconverted file, externally defined names in the reduced-version code file with original names that are used in the full-version code file, wherein the mapper utilizes an export file that was created during the pre-conversion step, and wherein external references found in the preconverted file are resolved and converted into symbolic external reference for every class file that is referenced by the reduced-version code file, and

wherein the final conversion step includes:

converting the preconverted file into the converted file by rearranging individual code sections, header sections and link information so that they comply with the pre-defined language syntax, wherein the converted file includes a converted class description and converted executable instructions that are respectively derived from the preconverted code section and the preconverted code description section in the preconverted file;

in response to determining that the converted file complies with the pre-defined language syntax, creating a second cryptographic signature for the converted file;

storing the second cryptographic signature on the smart card; and

in response to the first cryptographic signature matching the second cryptographic signature, storing the reduced-version code file on the smart card, wherein the first cryptographic signature and the second cryptographic signature match only if the original full-version code file, the reduced-version code file, the preconverted file, and the converted file all comply with the pre-defined language syntax.

24. (new) The computer-implemented method of claim 23, wherein the converted file contains a different class file than the original full-version code file, but wherein the converted file and the original full-version code file are semantically identical according to syntax rules set by the pre-defined language syntax.

25. (new) The computer-implemented method of claim 24, wherein creating and comparing the first cryptographic signature and the second cryptographic signature are performed outside the smart card.

26. (new) A tangible computer-readable medium on which is stored a computer program code, the computer program code comprising computer executable instructions configured for verifying that a reduced set of code adheres to a pre-defined language syntax before being loaded onto a smart card, and wherein the computer executable instructions, when executed, perform the steps of:

determining if an original full-version code file complies with a pre-defined language syntax, wherein the original full-version code file is capable of executing all instructions and utilizing all features found in a full version of a programming language;

in response to determining that the original full-version code file complies with the pre-defined language syntax, creating a first cryptographic signature for the original full-version code file;

storing the first cryptographic signature on a smart card, wherein the smart card comprises a limited-capacity embedded microcontroller;

converting the original full-version code file into a reduced-version code file, wherein the reduced-version code file is capable of executing only a limited subset of instructions and features found in the full version of the programming language;

converting the reduced-version code file into a converted file, wherein the converted file is capable of utilizing all features found in the full version of the programming language, and wherein converting the reduced-version code file into the converted file is performed by a pre-conversion step, a mapping step, and a final conversion step,

wherein the pre-conversion step includes:

converting the reduced-version code file into a preconverted file, wherein the preconverted file utilizes the pre-defined language syntax, and wherein the preconverted file includes a preconverted code section and a preconverted code description section, and

wherein the mapping step includes:

utilizing a mapper to replace, in the preconverted file, externally defined names in the reduced-version code file with original names that are used in the full-version code file, wherein the mapper utilizes an export file that was created during the pre-conversion step, and wherein external references found in the preconverted file are resolved and converted into symbolic external reference for every class file that is referenced by the reduced-version code file, and

wherein the final conversion step includes:

converting the preconverted file into the converted file by rearranging individual code sections, header sections and link information so that they comply with the pre-defined language syntax, wherein the converted file includes a converted class description and converted executable instructions that are respectively derived from the preconverted code section and the preconverted code description section in the preconverted file;

in response to determining that the converted file complies with the pre-defined language syntax, creating a second cryptographic signature for the converted file;

storing the second cryptographic signature on the smart card; and

in response to the first cryptographic signature matching the second cryptographic signature, storing the reduced-version code file on the smart card, wherein the first cryptographic signature and the second cryptographic signature match only if the original full-version code file, the reduced-version code file and the converted file comply with the pre-defined language syntax.

27. (new) The tangible computer-readable medium of claim 26, wherein the converted file contains a different class file than the original full-version code file, but wherein the converted file and the original full-version code file are semantically identical according to syntax rules set by the pre-defined language syntax.

28. (new) The tangible computer-readable medium of claim 27, wherein creating and comparing the first cryptographic signature and the second cryptographic signature are performed outside the smart card.

29. (new) A computer system comprising a tangible computer-readable medium on which is stored a computer program code, the computer program code comprising computer executable instructions configured for verifying that a reduced set of code adheres to a pre-defined language syntax before being loaded onto a smart card, and wherein the computer executable instructions, when executed, perform the steps of:

determining if an original full-version code file complies with a pre-defined language syntax, wherein the original full-version code file is capable of executing all instructions and utilizing all features found in a full version of a programming language;

in response to determining that the original full-version code file complies with the pre-defined language syntax, creating a first cryptographic signature for the original full-version code file;

storing the first cryptographic signature on a smart card, wherein the smart card comprises a limited-capacity embedded microcontroller;

converting the original full-version code file into a reduced-version code file, wherein the reduced-version code file is capable of executing only a limited subset of instructions and features found in the full version of the programming language;

converting the reduced-version code file into a converted file, wherein the converted file is capable of utilizing all features found in the full version of the programming language, and wherein converting the reduced-version code file into the converted file is performed by a pre-conversion step, a mapping step, and a final conversion step,

wherein the pre-conversion step includes:

converting the reduced-version code file into a preconverted file, wherein the preconverted file utilizes the pre-defined language syntax, and wherein the preconverted file includes a preconverted code section and a preconverted code description section, and

wherein the mapping step includes:

utilizing a mapper to replace, in the preconverted file, externally defined names in the reduced-version code file with original names that are used in the full-version code file, wherein the mapper utilizes an export file that was created during the pre-conversion step, and wherein external references found in the preconverted file are resolved and converted into symbolic external reference for every class file that is referenced by the reduced-version code file; and

wherein the final conversion step includes:

converting the preconverted file into the converted file by rearranging individual code sections, header sections and link information so that they comply with the pre-defined language syntax, wherein the converted file includes a converted class description and converted executable instructions that are respectively derived from the preconverted code section and the preconverted code description section in the preconverted file;

in response to determining that the converted file complies with the pre-defined language syntax, creating a second cryptographic signature for the converted file;

storing the second cryptographic signature on the smart card; and

in response to the first cryptographic signature matching the second cryptographic signature, storing the reduced-version code file on the smart card, wherein the first cryptographic signature and the second cryptographic signature match only if the original full-version code file, the reduced-version code file and the converted file comply with the pre-defined language syntax.

30. (new) The computer system of claim 29, wherein the converted file contains a different class file than the original full-version code file, but wherein the converted file and the original full-version code file are semantically identical according to syntax rules set by the pre-defined language syntax.

31. (new) The computer system of claim 30, wherein creating and comparing the first cryptographic signature and the second cryptographic signature are performed outside the smart card.